Application Development Tools

INPUT®

APPLICATION DEVELOPMENT TOOLS

APPLICATION DEVELOPMENT TOOLS

ABSTRACT

This report is the third in a series on development tools for the Information Systems Program.

The report focuses on the competitive environment for tools, aids, and design methodologies. More specifically, it analyzes the market requirements for end-user systems.

This report contains 53 pages, including 11 exhibits.

APPLICATION DEVELOPMENT TOOLS

CONTENTS

EXHIBITS

I INTRODUCTION

- This is the third report in a series which includes:
  - Data Base Management Systems.
  - Fourth Generation Languages.
- These reports form an integrated set and should be viewed as such.
  - The first report (Data Base Management Systems) emphasized the importance of DBMS in IBM's software strategy and pointed out important systems considerations based on projected hardware/software technological developments.
  - The second report (Fourth Generation Languages) emphasized the need for structure in software market analysis and presented a set of systems categories which are useful in establishing a frame of reference when considering the use of application development tools.
- This report will concentrate on the competitive environment in the overall market for tools, aids, and techniques to improve productivity in applications development. In doing so, INPUT will emphasize user requirements necessary to make effective use of current products, not the products themselves. It is our opinion that we have currently gone past the point where the "solutions" can be used to define the problem.

- Since this report series was specified, a rather nasty controversy has developed around the definition of a relational data base system. It all started when E.F. Codd, the primary inventor of the relational model, published an expanded version of definitions contained in his 1981 ACM Turing Award Lecture.

  - It is doubtful that anyone disputed Codd's right to define his creation after the Turing Award Lecture (or any of his other technical publications). In fact, INPUT has specifically stated that "Since Codd is so closely identified with the relational model, it seems only reasonable to accept his definitions of the relational model and what constitutes a relational data base system." (Relational Data Base Development, INPUT, August 1983). However, this time he was published in Computerworld, which also carries extensive advertising for software products including "relational" data base systems.

  - Unfortunately, the term "relational" has been applied to many successful (and unsuccessful) products rather indiscriminately and some vendors saw fit to take issue with Codd's definitions. This is unfortunate since the real value of the relational model rests as much with its solid theoretical foundation as with its external characteristics.

  - In Codd's response to some of the criticism of his articles, he emphasized the need for precise definitions for software product evaluation and market analysis, for not only DBMSs but for languages as well. Specifically, he states:

    . "There is no fourth generation language definition worth its salt, let alone any theoretical foundation. James Martin's purported definition fails to mention what capabilities a fourth generation language should have. . ."

- "Thus, any vendor can claim to provide a product that supports a fourth generation language, and there is no basis for checking or challenging such a claim."

- This is precisely the point INPUT made in its report on fourth generation languages--the term has no meaning in the marketplace, and until we clean up terminology and definitions, it is meaningless to expend great effort in product evaluation.

- This lack of structure at the most fundamental level is precisely the reason INPUT saw fit to combine these three reports. DBMSs, FGLs, and ADTs (application development tools) do not have definitions and are all competing for the same market, which can be roughly defined as "the market for ways to improve productivity in the systems (applications) development process."

II EXECUTIVE SUMMARY

- This executive summary is designed in a presentation format in order to:
  - Help the busy reader quickly review key research findings.
  - Provide an executive presentation and script that facilitates group communications.
- The key points of the entire report are summarized in Exhibits II-1 through II-5. On the left-hand page facing each exhibit is a script explaining the exhibit's contents.

A. PRODUCTIVITY (PERFORMANCE) LEVELS

- The impact of application development tools and the resulting applications systems must be measured at four performance levels.
  - Hardware/software costs may be increasing more rapidly than people costs.
  - The costs of both humans and machines must be considered in evaluating improved individual productivity.
  - Work unit networks may create more information, but quality may suffer.
  - The bottom line is whether the organization truly benefits, and it may not.
- Getting things done faster may result in negative performance impacts at all four levels because there are residual costs associated with computer/communications systems.

INPUT®

PRODUCTIVITY (Performance) LEVELS

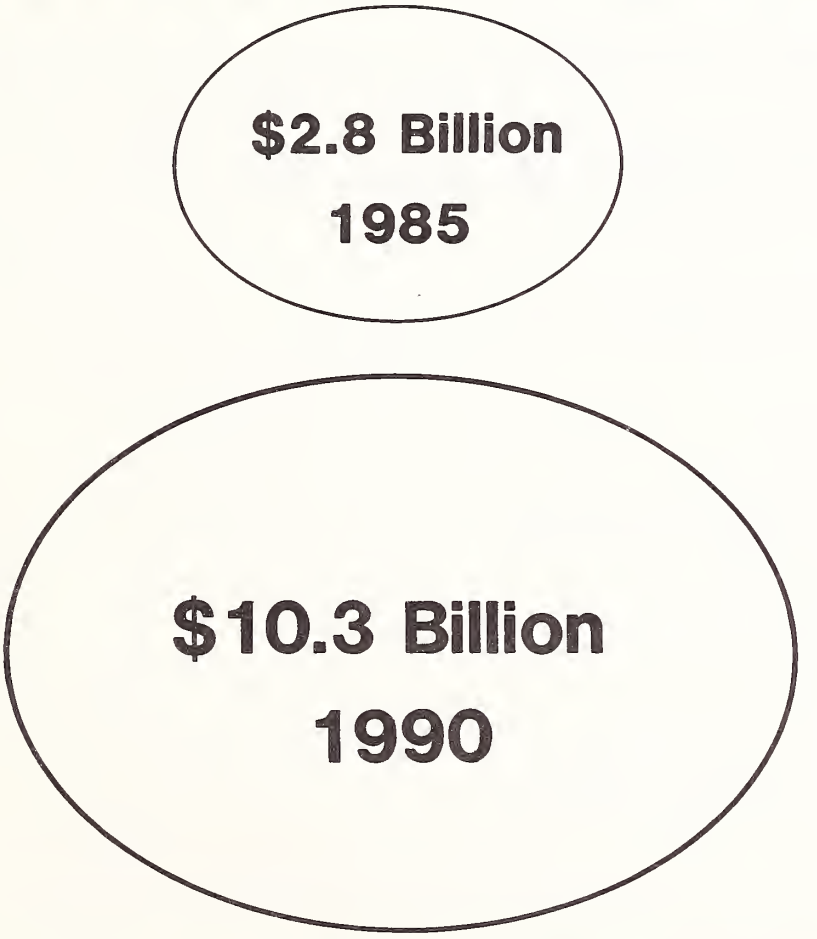
- Hardware/Software
- Human/Machine Dyad
- Work Unit Network
- Institutional

B. EXPANDED ADT MARKETS EQUAL EXPANDING I.S. COSTS

- INPUT projects the application development tools (ADT) market will expand rapidly between now and 1990.
- The use of ADTs increases the cost of the hardware/software performance level beyond the cost of the specific ADT itself.
- This will place even higher demands for improved performance at the other performance levels.
- The product of an information system is information, and there is no assurance that quality will improve with increased hardware/software expenditure.
- IS management has the responsibility to see that quality is maintained and improved. Control over the development process must be maintained.

INPUT®

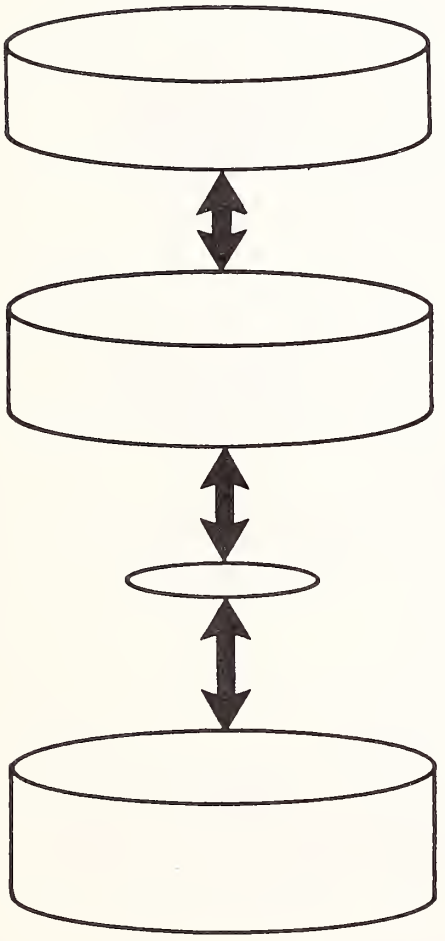
EXPANDING ADT MARKETS = EXPANDING I.S. COSTS



C. DATA ARE FUNDAMENTAL

- Information quality cannot be good unless data quality is good.
- Multiple data sources compromise data quality (integrity and syncronization), security, costs, and, as a result, information quality.
- Multiple DBMSs and languages complicate the problem of information quality.
- IBM's software strategy dictates how data will be distributed and accessed, and it is essentially a multiple operating system (VM, MVS, UNIX, etc.), DBMS (IMS, DB2, etc.), language (SQL, Intellect, PL/I, COBOL, etc.), and LAN strategy with many open questions.

DATA ARE FUNDAMENTAL



Mainframe Data Bases

Micro-Mainframe Links

Departmental Data Bases

Personal Data Bases

Public Networks

Public Data Bases

D. A DBMS STRATEGY

- INPUT believes relational DBMSs are essential in both IBM's strategy and in the development environment.

- Relational DBMSs must be analyzed to determine the degree of "purity" required before being used as the essential linkage among systems in the network hierarchy.

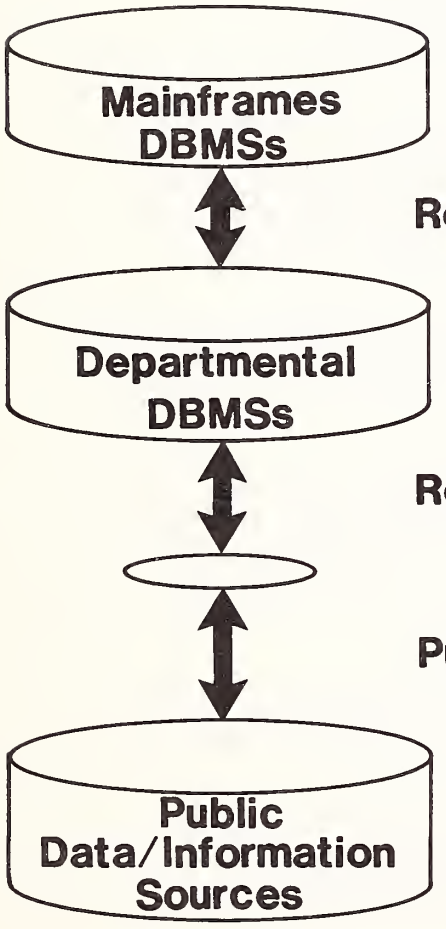
- Performance of hardware/software will continue to be a major concern with relational systems at the mainframe level, but the "production" DBMS must provide a convenient link to relational systems at lower levels in the hierarchy.

- Public data/information sources should be evaluated and used to complement and supplement internal resources.

- Control must be exercised over these external resources in terms of both cost and quality.

INPUT®

A DBMS STRATEGY



Standard + Relational

Relational Tables

Standardize on "Relational Type" DBMS

Relational Tables

Standardize on "Relational Type" DBMS

Public Networks

Evaluate on Quality & Cost

E. ESTABLISH A QUALITY ASSURANCE PROGRAM

- At the very least, a thorough analysis of residual costs should be made of various development tools and strategies.

- Hardware/software monitors should be employed throughout the network to assist in tuning performance, refining cost recovery, and as feedback to residual cost analysis.

- Concentrate research and systems effort on the analysis and control of data/information/knowledge quality, including such anticipated phenomenon as sharply increased entropy (the natural tendency to chaos).

- Incorporate network and data base modeling tools and/or services to predict computer/communications network performance.

- Establish appropriate standards and procedures.

INPUT®

ESTABLISH A QUALITY ASSURANCE PROGRAM

- Analytical Cost Analysis
- Hardware/Software Performance Monitors
- Data/Information/Knowledge Base Quality Control
- Network Modeling & Management
- Data Base Modeling

III APPLICATION DEVELOPMENT TOOLS - MEANS OR END?

A. MARKET REQUIREMENTS

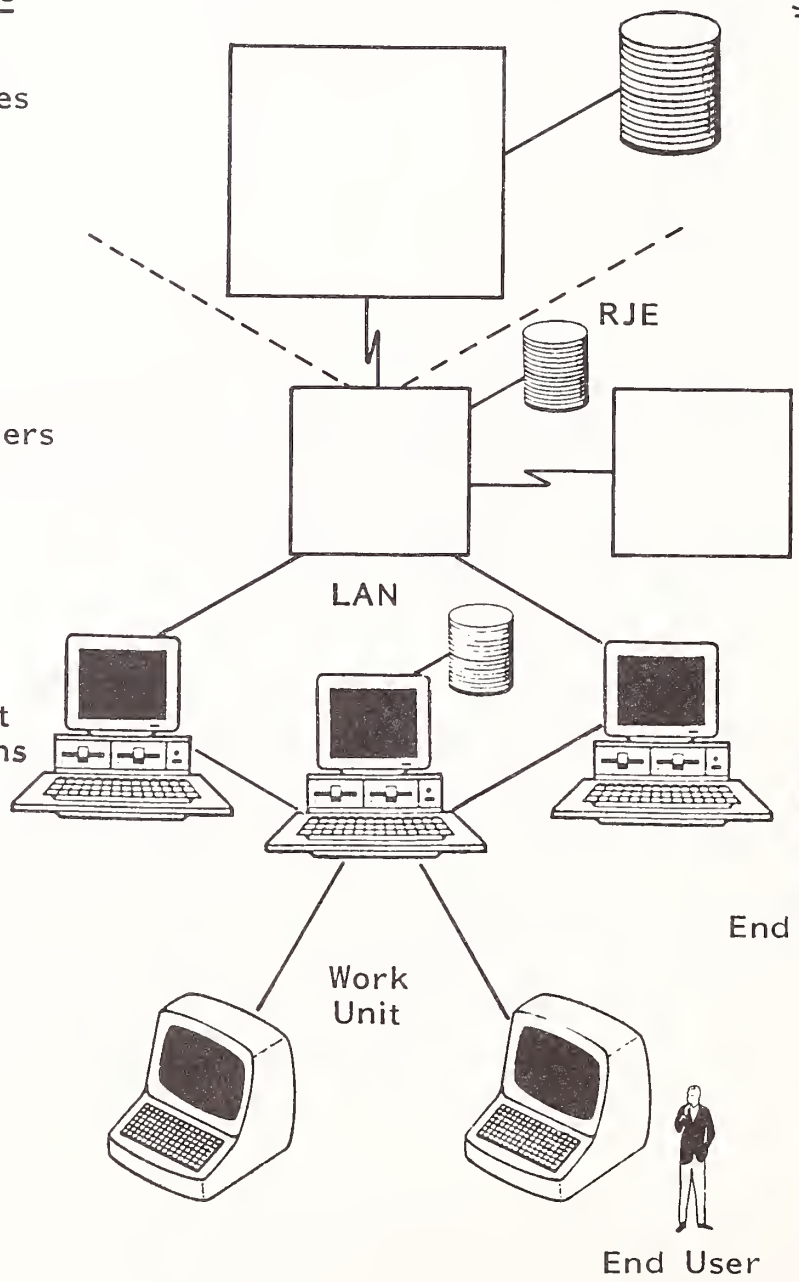
- INPUT starts with the assumption that the market for application development tools is determined by the applications which will be developed with those tools rather than the other way around (where the tools determine the application). Sometimes this does not seem to be the case; our industry is becoming noted for changing problems to fit solutions.

- Having made that assumption, it is possible to gain considerable insight into market requirements by relatively simple analysis of what users are saying, which is essentially this: "I want to be able to sit at an intelligent workstation (or have my employees sit at their workstations) and have ready access to all of the data and processing power of the network without regard for where the work is actually done." In other words, the applications and their necessary data will be distributed over a computer/communications network and will flow over that network freely, with only minimal navigational direction from the end user.

- Using INPUT's network hierarchy systems category (see Appendix A of Fourth Generation Languages for a list of systems categories), it is possible to visualize the market requirements quite clearly (see Exhibit III-1).

EXHIBIT III-1

APPLICATION STRUCTURE



NETWORK HIERARCHY Level	Hardware	PRIMARY SYSTEMS TYPES
I	Large Mainframes	Batch
II	Minicomputers	Interactive Transaction
III	Intelligent Workstations	Decision Support Expert Systems
IV	"Dumb" Terminals	Transaction Real Time
V	Mobile Terminals (To Level IIs)	Interactive Transaction Real Time

- INPUT has assumed that there is a "natural" or "proper" network hierarchy associated with computer/communications networks. This proper hierarchy is dictated by its cost effectiveness and determines the appropriate functions (applications) at the various levels. This hierarchy was first presented by INPUT 10 years ago, and it has remained relatively unchanged except for terminology.

- The functions originally assigned at Level I were as follows:

  . Heavy computation.

  . Transaction processing against large data bases.

  . RJE replacement of standalone batch systems.

- The original functions at Level II were as follows:

  . Network control.

  . Scientific timesharing.

  . Program development and maintenance.

  . Simple transaction processing.

- The original functions at Level III were:

  . Collection, editing, and display of data (and information).

  . Control of Level IV terminals.

- The original functions at Level IV were:

  . Data entry and display (including printing).

  . Sensing and control devices.

- Level V mobile (portable) terminals were not shown in the original hierarchy, but they were mentioned in the body of the report.

- While it is obvious that the network depicted falls conveniently into the established hierarchy and today's terminology, there might be some dispute when the systems type systems category is assigned at various levels. It is INPUT's position that any discrepancies are primarily the result of IBM's strategy (for example, running UNIX on mainframes is in keeping with IBM's reluctance to distribute interactive processing to its proper level). However, from the point of view of end users at Levels III, IV, and V, usage (applications) patterns will soon develop which support this distribution.

  - It is all well and good for users to state they do not want to be concerned about where data are, or where processing is done, but there is one thing they are concerned about and that is cost. As users become directly involved in the operations (and development) of their systems they will be in a much better position to understand relative costs. It is INPUT's opinion that this increased user awareness will encourage, if not force, the cost-effective distribution of functions and systems types into a "proper" hierarchical network through usage patterns.

  - Perhaps the simplest illustration of the natural tendency toward cost-effective use is found with the use of today's public information networks such as The Source and CompuServ. Early usage patterns may find users composing correspondence and browsing through information on-line, but it does not take many monthly bills to convince most of them that file transfer makes a lot more sense.

- It will not take long for users to become aware of the cost of constantly enquiring into corporate data bases, and there will be a natural tendency to upload and download files and data base subsets based on both response time and cost. A few JOINs and SELECTs against large DB2 data bases on the host computer will be more than enough to convince most users that there must be a better way. (This statement is made with all due respect for Dr. Codd, whose relational model will become the bedrock upon which these distributed systems will be built.)

- Then, of course, with IBM's multiple data base strategy, the extract runs against IMS data bases, VSAM files, and sequential files to create relational tables are going to be far from interactive. A user requesting data from an archival tape file may not have to know where the data resides (or on what media), but the application should be designed to tell him when he can expect to receive it (hours, days, or whatever) and how much it will cost.

- The current, popular emphasis among IS management is upon "connectivity," and there is an awareness that future systems will be based on communications. Unfortunately, both users and vendors associate anything connected to a computer/communications network as being interactive and that is not the case at all. The telephone system has normally been interactive because it was necessary for both parties to be connected at the same time, but it must be recognized that the bulk of communications still takes place on paper and the U.S. Postal Service is also a communications network. It may be possible to reduce the massive paper output of computer systems (most of which has been distributed through internal or external mail), but the protocols between Levels I and II and II and III (or micro-to-mainframe) are going to look more like 2780 batch processing than they are interactive timesharing.

- Both IS management and many vendors prefer to ignore batch processing, but it is not going to go away. If the term has become an anathema, perhaps we can refer to communications between various network modes as being in "blast" or "flash" mode (we are good at changing terminology and not concepts), but the requirement for tools and aids to develop flexible batch applications remains regardless of whether anyone wants to recognize it.

- Systems designers and applications developers need tools which will:

  - Assist them in accommodating a variety of user languages at Levels III, IV, and V and a variety of data base systems at all levels in the hierarchy.

  - Provide facilities for identifying sources of data/information/knowledge and for incorporating these data/information/knowledge into the applications systems being developed.

  - Guide them in how data/information/knowledge should be distributed over the network hierarchy in order to achieve balanced productivity improvement as defined by INPUT's performance systems category, which includes the following levels:

    . Hardware/software.

    . Human/machine dyad.

    . Work unit network.

    . Institutional.

  - Help them in determining which tools to use in order to achieve balanced performance improvement (productivity) in terms of the systems requirements systems category, which includes the following subcategories:

  - . High/low transaction rates.
  - . High/low processing requirements.
  - . Large/small data base size.
  - . High/low functionality.
  - . Many/few decision rules.
  - . High/low responsiveness.

- Facilitate the development of quality assurance programs for the applications systems being developed, including the subcategories contained under INPUT's quality systems category:
  - . Objectives.
  - . Data/information/knowledge.
  - . Auditability.
  - . Measurement.
  - . Feedback loops.
  - . Validity/reliability/predictability (of achieving objectives).
  - . Security/privacy.
- Provide flexibility and facilitate change in all of the above.

- Vague requirements definitions such as "connectivity," LANs, and micro-mainframes links and universal solutions such as relational DBMSs, 4GLs, "information engineering," and data-driven prototyping all become part of the problem when the systems analyst and development manager are confronted with developing quality systems (much less integrating the hodgepodge of prototypes, expert systems, and communications systems being developed). The chaos which currently exists in the systems development process (and among the "experts" in the industry) is the direct result of improperly applied tools (both hardware and software). What are needed are tools, techniques, and approaches which facilitate, direct, and/or force intelligent application of many of the tools already available.

- Therefore, if the requirements above do not correspond with your particular "solution," appear complex or impossible, and are not being specifically articulated by systems personnel, it is not surprising. Both the problem and some more detailed requirements were presented in New Opportunities for Software Productivity Improvements, INPUT, 1984, and events of the last year have only confirmed the findings of that report. The specific recommendations of that report will be summarized later.

B. CURRENT PRODUCTS

- There are a great variety of productivity tools available to address specific aspects of the requirements outlined above. INPUT classified these tools into some general categories by systems development phase in a 1983 Vendor Watch Report on Software Productivity Tools: Update and Outlook (see Exhibit III-2). The report then attempted some additional clarification by regrouping the tools into "pre-implementation," "implementation", and "revolutionary" categories.

EXHIBIT III-2

CLASSIFICATION OF SYSTEM PRODUCTIVITY TOOLS (SPTs)

STP	EXAMPLES	SYSTEMS DEVELOPMENT PHASE ADDRESSED		
		Requirement Definitions	Design	Implementation
Artificial Intelligence	LISP, SMALL TALK	X	X	X
Data Dictionary	Datadictionary (ADR) DB/DC (IBM) UCC-10 (UCC)	X	X	X
Data-Driven Prototyping	PDM-80	X	X	X
Design Methodologies	Structural Design (De Merco) Structural Analysis	X	X	
Information Planning	Business Systems Planning Information Modeling	X		
Modeling/Non-procedural Languages	Focus, Express, Easytrieve	X	X	X
Programming Aids	Program Utilities (CAPEX) Structural Programming			X
Visual Programming	MAPPER, VisiCalc		X	X

- Pre-implementation (requirements definition/design) tools were listed as including the following:

  . Business or information systems planning (e.g., IBM's BSP).

  . Data gathering/analysis techniques (e.g., information modeling).

  . Structured analysis/design (e.g., DeMarco, Yourdon, SofTech, HIPO, and PRIDE).

  . DBMSs.

  . Software aided (e.g., DDI - J. Martin, DSSD - K. Orr).

  . Application prototyping.

  . Data dictionaries.

- Implementation tools were listed as follows:

  . Structured programming (e.g., SPF).

  . Program code generators.

  . Higher level retrieval languages (e.g., DYL 280, Easytrieve).

  . Fourth generation languages (e.g., Focus, INTELLECT).

  . DBMSs.

  . Programming utilities (e.g., Capex, Optimizer).

  . Systems management aids (e.g., JARS).

  . Telecommunications monitors (e.g., CICS).

- Revolutionary techniques were described as spanning both the pre-implementation and implementation phases and were listed as follows:

  . Visual programming (e.g., MAPPER, VisiCalc).

  . Data-driven prototyping (e.g., PDM-80 from DACOM).

  . Artificial intelligence (e.g., exploratory programming: LISP, SMALLTALK).

- INPUT then stated: "As depicted in the chart, the earliest SPTs (system productivity tools) were intended to support programmers. These tools increased programming productivity, but they did not increase system development productivity. IS management concluded that programming the wrong system faster would not solve the problem. New tools were developed to better define requirements. Still more tools are being developed to cover all phases of systems development life cycles, starting with systems planning and needs analysis and continuing through performance monitoring."

The above listings made no pretense of being comprehensive, but they did point out the wide variety of products competing in the market for software productivity tools. More importantly, they provided the insight to identify the orientation of the major competitive thrusts in the marketplace. Fundamentally, there are thrusts coming from three directions, and they are all directed at penetrating the same markets (the ones which cover all phases of the systems development life cycle):

- There are competitors coming basically from a DBMS orientation.

- There are those whose primary emphasis has been language oriented (as manifested by 4GLs).

- And there are the new brand of "PC jockeys" who have started from standalone personal computers ("visual programming" really originated here) and now find themselves hooked into the network hierarchy at Levels III, IV, and V.

- Regardless of the original orientation, all tools are confronted with integration problems as their use is extended into markets where others have established early penetration. Each of the three major thrusts have something to learn from the others. For example, developers of DBMSs and 4GLs may think they understand something about "ease of use" until they encounter the new user or those weaned on PC software. Integrated PC software vendors may think they understand DBMSs until they run into the data base integrity and security problems associated with shared use and distributed data bases. Then, overlaying the whole problem is the fact that the targets keep changing. Consider the following quote which was recently published.

  - "The industry has been very slow to recognize how quickly people become power users." (Attributed to Richard Rabins, president of Alpha Software.)

  - There is probably some element of truth in the above, but it does not help much in determining the types of productivity tools which are required.

- Of course, there are those who recognize the complexity of the problem and the deficiencies of putting too much dependency on past solutions. New, comprehensive solutions (such as data-driven prototyping) frequently have merit, but have great difficulty becoming accepted for the following reasons:

  - Users have been exposed to so many solutions and promises that they will not take the time to consider (much less understand) a new approach or tool.

- The developers of the new tool cannot find qualified personnel to market, sell, or install their product.

- The cost of launching new products (even software products) is increasing substantially as there are more and more announcements of new "solutions" and established vendors enhance their products and increase their customer base.

- Venture capitalists have been burned so often recently that they are extremely reluctant to provide the funds to launch a new software product.

- All of the above lead INPUT to believe that the primary competition will center around established vendors from the three primary product areas (DBMSs, 4GLs, and PC-oriented) competing against each other on the other guy's turf. In addition, only the well established (or those with a deep pocketed patron) have a chance of surviving in today's complex, disillusioned marketplace.

C. COMPETITIVE ENVIRONMENT

- Before discussing the competitive environment for application development tools, it is important to emphasize INPUT's analysis of the overall software marketplace. This analysis was summarized quite succinctly in Software News, September 1985 ("Software News' Top 50 Independent Software Vendors" by Peter Cunningham and Bonnie Digrius of INPUT).

  - "Increasing applications and systems software integration into a single product offering. Thus, the most successful products of the last half of the decade will have almost as much value added from systems software components as they will from applications software parts.

- "For example, the full value of a general ledger system will be as much due to integrated systems software components such as DBMS, micro-mainframe links, and 4GL (fourth generation languages) as it is to the basic accounting functions."

- "Emergence of a true distributed data processing environment (DDP). This new, more complex universe with its multi-layered processing and data base locations is causing product obsolescence. (Bad news for vendors with older product offerings and/or limited resources, but multiple new opportunities for major new product offerings.)"

- Then INPUT made the following observations relating to competitive structure:

  . Fortune 1000-type firms are becoming more aggressive marketers of software products. (For example, McGraw Hill and Continental Telecom.)

  . IBM is becoming more aggressive in pricing, developing, and making joint marketing agreements for software products for all sizes of computers in all major markets.

  . Non-software information services vendors (e.g., Martin Marietta Data Systems and Dun & Bradstreet) are expanding beyond their traditional offerings to include software products as part of their total offering.

- INPUT concluded that there were many opportunities for "innovative forward-looking vendors. . .willing to make major investments in quality management as well as marketing and technical resources."

- Remembering the ever-ominous presence of IBM, let's take a look at the major independent competitors from the perspective established in this series of reports (see Exhibit III-3).

  - There are four major vendors coming from a DBMS orientation with combined revenues of nearly $400 million in 1984. All four are currently giving attention to both 4GLs and PC links of some kind.

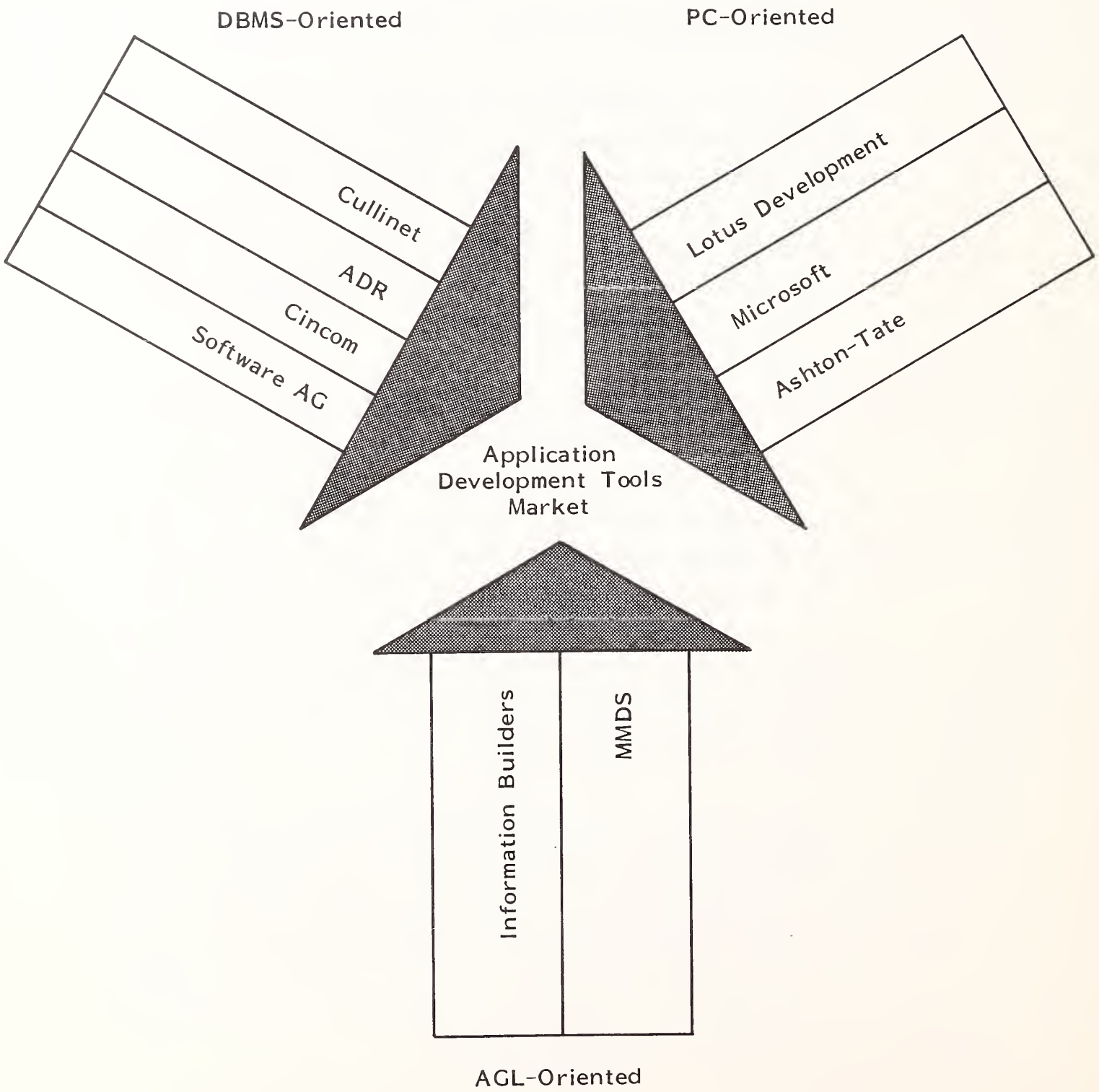
    . Cullinet is the largest independent software vendor in the world with $167 million in revenue. It offers not only IDMS and IDMS/R (a less than relational DBMS according to Dr. Codd), but also ADS/OnLine (which is billed as being "more than a fourth generation language"). After limited acceptance of its Goldengate integrated package for micros, Cullinet developed Infogate, a link to multiple micro-based systems.

    . Applied Data Research Inc. (ADR), with software revenue of about $128 million, provides Datacom/DB (which did not fare any better than IDMS/R when subjected to Dr. Codd's scrutiny) as its base product along with Ideal (a successful 4GL despite some adverse publicity), PC Datacom 2.0 (a micro-mainframe link to Datacom/DB), and PC Peer (a five-function integrated software package for the IBM PC). (In late November, ADR was acquired by Ameritech, a Bell system spinoff with $8 billion in revenue and $1 billion in profits, which should make its fellow independent competitors pause.)

    . Cincom Systems, Inc., with $53 million in U.S. revenue ($90 million worldwide), shifted emphasis from the venerable TOTAL to an integrated set of products including TIS (a reported relational DBMS which has not been subjected to Dr. Codd's scrutiny in public), Mantis (a fourth generation language), Intellinet Query (a network management system), and PC Contact (a micro-mainframe link).

EXHIBIT III-3

WHERE ADT COMPETITORS ARE COMING FROM



- Software AG Systems Group, Inc., with $47 million in U.S. revenues, rounds out the DBMS-oriented competitors. Having started with Adabas, Natural (a fourth generation language) has been added along with Natural Connection (a micro-mainframe link between Adabas on the host and several micro packages).

- The three "whiz kids" of the PC-oriented competitors have revenue approximately equal to the four older DBMS-oriented firms (over $300 million). However, they come from different orientations and their future directions are less clear except to state that all must face the inevitability of communicating with other levels in the network hierarchy. Application development tools suited for early standalone PC applications are not going to survive as IBM moves from the SNA/DDP strategic period toward the electronic office period. Fortunately, all three seem to be maturing rapidly based on their early experience with integrated packages and micro-mainframe links.

- Lotus Development Corporation, with over $200 million in revenue, is the king of the integrated business packages with 1-2-3 (spreadsheet, data base, graphics). 1-2-3 has an installed base of over 1 million copies, but the follow-on package (Symphony) "only" sold 100,000 copies in 1984 and customers are beginning to resist the cost of new releases of 1-2-3.

- Microsoft, starting with an implementation of Basic for micro-computers, has grown to a $123 million company, extending into operating systems by providing PC-DOS to IBM and Xenix for the UNIX market. Close affiliation with IBM practically assures Microsoft's success during the SNA/DDP period, but it does not mean IBM will share very much of the electronic office market with its little pal.

. Ashton-Tate had 1984 revenue of $83 million and comes from a DBMS orientation with its dBASE II and III products, but attempts to compete against Lotus' Symphony with Framework have been "disappointing" with only 45,000 copies shipped in 1984.

- The early non-procedural languages (Ramis and Focus) have not experienced the type of growth exhibited by the DBMS and PC-oriented competitors (even after receiving the catchy description of fourth generation languages from James Martin). The combined U.S. software revenue of the two companies--Information Builders (Focus) and Martin Marietta Data Systems (which acquired Mathematica, the developer of Ramis)--was only $69 million in 1983. The conclusion is simple--it is easier to add a 4GL to an established DBMS than it is to add a comprehensive DBMS to an established language (not surprising). However, both of the 4GL-oriented competitors have a loyal customer base and other advantages.

. Information Builders, Inc. had $38 million in revenue in 1984, concentrating on decision support systems (DSS) with Focus, but it has introduced PC/Focus (designed for the IBM, TI, and Wang PCs), Foctalk, and Foccalc (a micro-mainframe link and a spreadsheet) and is porting micro functions to the mainframe as well as providing 4GL capability--a reasonably intelligent strategy in today's environment.

. Martin Marietta Data Systems (MMDS) had $31 million in software revenue in 1984 and Ramis contributed approximately 60% of this; however, MMDS is bringing together Ramis with UFO (an applications development system from Oxford Software Corporation, which is also part of MMDS) and this shows some appreciation for the fact that there is no magic solution to the productivity problem.

- In addition to those companies which seem to be directing their strategies toward direct competition in the projected market for application development tools, there are those who are more or less on the periphery with specific products and/or questionable direction (see Exhibit III-4). We will only comment on a few of them.

  - Informatics was the subject of a takeover by Sterling Software, Inc. It is INPUT's opinion that the distraction occurred at a critical time, otherwise Informatics would have been listed with the major competitors.

  - SAS is good at what it does and it has been astute enough to acquire a DBMS (System 2000 from Intel), but it has been going through some growing pains and some of the technical personnel have wandered off to do their own things. With careful planning, good management, and a little luck, SAS could outflank current competitors by providing integrated data/information/knowledge bases up and down the hierarchy.

  - Candle Corporation specializes in performance monitors for IBM systems software and predicting performance (and controlling) at the hardware/software performance level is a key element in the development of quality systems. But, performance measurement tools have traditionally been the concern of operations (in other words, they are employed after the fact) and today's DSD environment is based on studied disdain for the hardware/software performance level.

  - CGA Computer Inc. provides a security package (Top Secret) and security will receive renewed attention during the SNA/DDP strategic period, but today's tools do not necessarily work in tomorrow's environment and IBM has an inside track on comprehensive security systems.

EXHIBIT III-4

FRINGE COMPETITORS

COMPANY	REVENUE ($ Millions)	PRODUCTS
Computer Associates International	$81	Performance Improvement (Apex, Optimizer)
Informatics (Sterling)	74	Programmer Aids (4GL) (Mark Series)
Pansophic Systems Inc.	45	Application Development Tools Operations Management (Panvalet, Easy Third)
SAS Institute	45	Information Management (SAS, S2000)
Candle Corp.	40	Performance Monitoring (MVS, IMS, VM)
CGA Computer Inc.	20	Security (Top Secret)
Boole & Babbage	17	Performance, Production and Security

- In this series of reports, INPUT has made clear that the market for application development tools is heavily dependent upon a solid data base foundation and that those vendors with a DBMS product orientation will be the primary beneficiaries of that market growth. Those with a language orientation will lose market share to the DBMS-oriented vendors (market forecasts for FGLs in Market Analysis: Fourth Generation Languages were adjusted to reflect this shift) and vendors of other application development tools, such as those operating around the fringe of the ADT market, will represent a relatively small percentage of the total market.

- It has also been pointed out that IBM's primary emphasis during the SNA/DDP period will be upon operating systems and DBMSs as means of establishing and maintaining control of the emerging distributed processing environment. In addition, to the degree that IBM concentrates on DBMS, the market for advanced language development will be more open to other competitors. IBM's attention to the "other" application development tools will be directly related to the strategic importance of the particular tool. For example, it can be anticipated that IBM will be less than interested in tools to monitor and predict performance but will consider tools to develop highly secure systems to be of strategic importance.

- Since DBMS is so important to the ADT market, we would like to review some of the thinking (or lack thereof) which has surfaced in various publications since the DBMS report of this series was published a few months ago. Essentially, there seems to be a school of thought developing which says the following:

  - The relational "craze" will give way to more "robust" (which seems to be the latest craze in terminology) systems which look more like the ANSI three-schema architecture. Therefore, pure relational systems will never achieve substantial market penetration.

- Distributed data bases have substantial unsolved problems (agreed) and the need for them is limited; besides, centralized processing power has been able to keep up with the demands being made on central hosts (or clusters of hosts) at a reasonable cost.

- Data base machines will continue to develop very slowly and IBM will not be putting essential DBMS (and/or operating systems) functions in microcode because non-IBM DBMS software has become too important to IBM's major customers.

- IBM does not have its data base act together and is threatened with substantial loss of market share.

- INPUT's comments on these conclusions are as follows:

  - The relational "craze" was thinking the entire world would ever be relational to begin with, but it now appears that those who embraced the "relational" cause in order to belabor IBM about its dual DBMS approach at the time DB2 was announced are now backing off. It is probable that this is the direct result of Dr. Codd taking dead aim at the relational clay pigeons which have proliferated in the marketplace (or perhaps it is just a question of some of the experts beginning to understand what a relational DBMS is).

  - Unfortunately, this sudden 180 degree reversal comes at precisely the time when the relational model is most important, and that is when distributed data bases begin to develop. Which brings us to the second point--whether there is a "true need" for distributed data bases or not, they are going to develop in the DSD environment regardless of whether or not the problems associated with them are solved. The stringent rules associated with Dr. Codd's relational DBMS definition become especially important in maintaining integrity, and the flexibility and ease of use of a relational DBMS become essential at Levels

II and III in the network hierarchy. A substantial portion of the projected DBMS market is going to be associated with those levels.

- It is INPUT's opinion that micro-mainframe links are going to place enormous processing demands on central host processors. This in turn will encourage (and even force) both the geographic and architectural distribution of processing (the term geographic distribution refers specifically to distributed data bases and the term architectural distribution connotes some type of data base machine). The same processing crunch will also encourage IBM to relieve the systems software overhead by putting more functions into microcode. Then the obsolete software DBMSs, both IBM's and others, will be left to die on the vine and continue to absorb any excess processing power which might have existed otherwise.

- In Market Analysis: Data Base Management Systems, INPUT stated that it did not believe that IBM was going to lose DBMS market share and the reasons for that conclusion. Events since that time have only tended to confirm this opinion, and it is probable that IBM will actually gain market share over the 1985-1986 timeframe.

- It has been customary to view IBM as being primarily a hardware peddler and there were those in IBM who seriously questioned IBM's decision to unbundle software nearly 20 years ago. (The usual question was, "Who would pay for it?") However, there are a few facts which discredit this point of view.

  - Everyone knows that IBM controls the mainframe market, and yet the top 10 mainframe competitors worldwide have 77% as much mainframe revenue as IBM but only derive 45% as much software revenue as IBM.

  - The top 10 independent software vendors in the U.S. only have 30% as much software revenue as IBM, and the top 50 only account for 62% of IBM's software sales.

- If software earnings figures were available for comparison, the dominance of IBM would be even more striking. In addition, the argument that IBM's software sales are primarily derived from "captive" systems software sales are not a sign of vulnerability in other areas--it merely means competitors' products (whether ADTs or applications) depend upon IBM for their very existence.

- There is a lot of whistling in the dark about the software competitive environment, but it does not obscure the fact that IBM is firmly in charge and those who do not recognize this fact are going to find they have been walking through the cemetary.

IV OPPORTUNITIES AND CHALLENGES

A. ADT FORECAST

- Market forecasts are not normally included in reports directed to the user community, but INPUT feels they are important in summarizing what we feel you are going to be doing over the next few years. As our forecasts become more refined, they will represent with increasing reliability technological progress in the computer/communications industry. Since the limiting factor on technological progress is the development of applications systems, the overall market for application development tools is a convenient bellwether to determine technological direction and progress.

- INPUT projects the total application development tools market to be $10.3 billion in 1990 (see Exhibit IV-1). This will be broken down as follows:

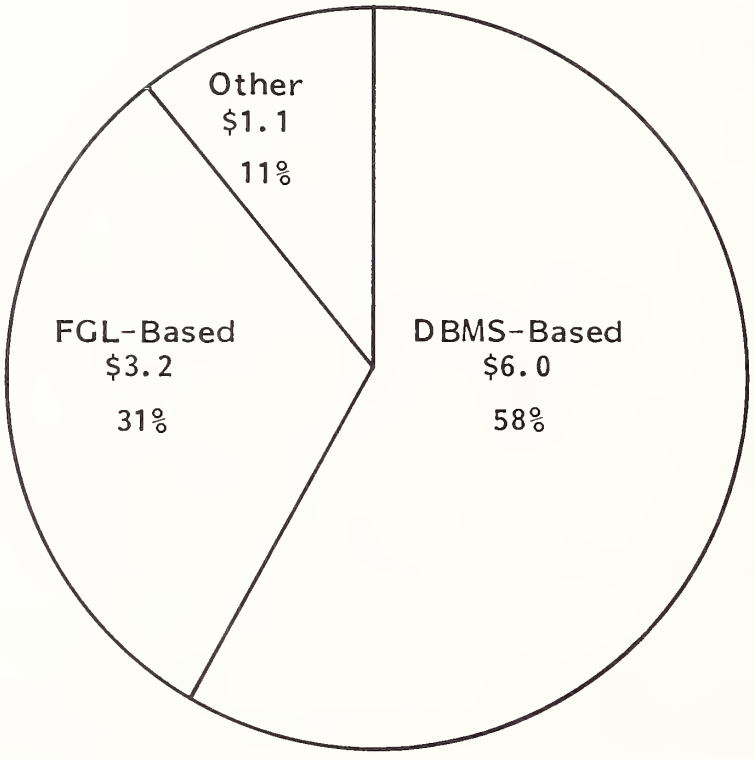
  - $6 billion of the market will develop from DBMS-based products.

  - $3.2 billion of the market will develop from language-based products (FGLs). This includes expert systems.

  - The remaining $1.1 billion is classified as "other," a category which includes both the specialized and highly advanced tools anticipated.

EXHIBIT IV-1

ADT FORECAST, 1990
($ Billions)



Total ADT Market: $10.3 Billion

- The primary distinguishing factor of the "other" category is that the tools are directed toward the problems which remain after more generalized tools such as DBMSs and 4GLs are applied, and it is anticipated that most such tools can be viewed as being either complementary or supplementary to the major categories of ADTs and to vendor-provided operating systems.

- Using various systems categories as specific frames of reference, it is not difficult to isolate these remaining problems and even anticipate how tools themselves contribute to these problems. For example:

  - INPUT's productivity hierarchy (pyramid) has always emphasized that "commitment to quality" is of primary importance in any productivity improvement program, and to the degree that tools and aids facilitate the development of "quick and dirty" systems, they can contribute to the problem. (This was the essential theme of New Opportunities for Software Productivity Improvement, INPUT 1984.)

  - Using data-driven systems development methodologies (prototyping) may address the early phases of the development/life cycle systems category (requirements, specifications, etc.) and an important level of the productivity hierarchy (end-user involvement, second only to commitment to quality in importance), but they can cause serious problems in some of the quality subsets such as:

    . Objectives.

    . Measurement.

    . Auditability.

    . Validity/reliability/predictability.

- Then, of course, the balancing of GST (general systems theory) directions varies over time in terms of emphasis. (IBM's highly centralized DBMS approach may be appropriate for the SNA/DDP period, but creates substantial problems--or opportunities--in the electronic office period.)

- Numerous other potential challenges for current application development tools can be isolated by reviewing the systems categories (Appendix A in another report, Fourth Generation Languages), but perhaps none is quite so evident as the performance category which has repeatedly been emphasized in this series of reports. It is essential that productivity be viewed in terms of performance at all four levels--hardware/software, human/machine dyad, work unit, and institutional. The emerging market for "other" application development tools will address specific performance levels and the balance across those levels.

B. CHAOS AMONG THE "SOLUTIONS"

- There is a natural tendency by both users and vendors to extend the use of specific tools beyond their intended or practical purpose. While a certain amount of this testing of product limits is both inevitable (and even desirable), the acceptance for application development tools is being adversely impacted by both specific and general misuse of existing tools. While a portion of this is clearly the responsibility of users who have had a persistent propensity to seek one simple solution to an extremely complex problem, vendors must accept a major share of responsibility for the general "buyer beware" atmosphere which pervades the marketplace.

- The claims which have been, and are being, made for various tools, aids, techniques, approaches, and methodologies are legend, going right back to that old granddaddy, COBOL. If 1% of the accumulated claims had actually been achieved, there would not be any productivity problem today. Both good

and bad tools suffer when a single hammer is advertised as being just right for driving tacks, nails, and pilings, or a Swiss army knife is used to build a house.

- There are reputable and knowledgable people (the two terms are not necessarily synonymous) who are stating that "anyone who speaks of productivity improvements on the order of 50-100% or more is a fraud." Contrast that with advertised and/or reported claims in the trade press or even technical journals, and then think of what it means in evaluating proffered solutions to the productivity problems in the systems development process.

  - Suppose a vendor has a product which actually improves performance by 100% over the development/life cycle from requirements definition to maintenance.

  - On the one hand, this level of performance improvement is not competitive with the claims of 500-1,000% (and even more) improvement which are routinely made (and even reported).

  - On the other hand, there will be documented cases (or rumors) where disastrous and catastrophic failures have resulted because even the best tool can be misused.

- A recent seminar announcement for DBMSs and 4GLs listed one-hour presentations from nearly 50 vendors and new products are being announced practically on a daily basis. It is impossible for any user (or consultant) to make any meaningful functional analysis of this vast array of products, much less any qualitative evaluation concerning performance. There is a natural tendency to look primarily to established vendors who advertise extensively and have a solid customer base from which they can reference sell. It will be extremely difficult for new vendors of conventional application development tools to get your attention, regardless of the quality of their product.

- This chaos in application development tools is the reason INPUT believes there is a substantial market for tools which will integrate and place boundaries on the use of already existing tools. Essentially, this is the "other" category which has been forecast.

C. CHAOS IN THE DEVELOPMENT PROCESS

- INPUT has emphasized the need for market structure (systems categories) because of the environmental chaos which has resulted from distributed systems development. The best way to illustrate the problem is to list four of the systems categories which have been proposed for structuring the market (see Exhibit IV-2).

  - In the DSD environment, the development structure (design, program, work unit organization, operational, and rigidity/flexibility) itself has become substantially more fluid.

    . Systems can be designed from the top down or be evolved from the bottom up.

    . Programs can be either structured or the most horrible hodge-podge imaginable. In addition, the unpredictable meanderings of exploratory programming (expert systems) will make algorithmic programs employing GOTOs appear to be relatively structured.

    . Work units established for development may be under highly centralized control (with established standards) or casual structures running both horizontally and vertically across established organizational boundaries.

EXHIBIT IV-2

CONFLICTS IN THE DSD ENVIRONMENT

- Development Structure
  - Conflicting Design Objectives and Implementation Methodologies
  - Conflicting Development Organizations and Work Units
  - Dynamic Source and Target Operating Environments
  - Prototype Versus Production Version
- Systems Type
  - Batch Versus Interactive
  - Decision Support Versus Expert
- Systems Requirement
  - Function Versus Performance
  - Data Base Size and Distribution
- User Set
  - "Dumb" Versus "Smart" Users

- Source and target operating environments can be at any combination of levels in the network hierarchy (mainframe, minicomputer, intelligent workstation, etc.) and subject to dynamic reallocation.

- The general objectives of the systems are also subject to change--the quick and dirty "pumpkin" is expected to change into an elegant "carriage" when the prototyping princess is ready to go to the production ball. However, the very flexibility which permitted a bumper crop of pumpkins may not meet the rigid expectations and standards of the grand event.

- The systems type category is already complicated enough by the different requirements of batch versus interactive and the gradations are becoming more complicated as finer distinctions are made and expert systems begin to appear.

- Systems requirements are getting complex (and of broader range) as more terminals go on-line, new analytical tools are used, data bases continue to grow astronomically in terms of size and content, and programs become more complex logically (more decision rules).

- In addition, the user set is ever expanding. It is our opinion that the inexperienced, first-time users include both the "dumb" and the "smart," and that among them are those who are more intelligent than either the sellers or developers of application development tools (or at least they have been paying the bills). These users will question the tools (spreadsheets, DBMSs, 4GLs, etc.) in terms of both function and cost.

- The point is that the application of ADTs is determined by the development environment (at present, the trend is toward distributed systems development), and this complex and changing environment can be roughly illustrated

by the systems categories in Exhibit IV-2. To expect any tool, or set of tools, to address all of the possible combinations of development structures, systems types, systems requirements, and user sets would be foolhardy and intelligent users and reputable vendors know this. However, the tendency to overextend the barriers of reason and good sense seems to be constantly with us; otherwise, the case study presented in Market Analysis: Fourth Generation Languages would never have occurred.

- Chaos in the development process is a direct result of the unrelenting search for the magic bullet to solve the productivity problem as manifested by the ever-growing backlog of user requests. Now, there are indications that a little legerdemain is going on with the backlog. By getting started on projects early (or perhaps prematurely) through the use of ADTs and information centers, the user requests are removed from the backlog and declared to be under development. The result of this is that the maintenance backlog is growing. If you find this to be alarming, you are in good company--it was the primary concern of a recent meeting of the Quality Assurance Institute.

- There is also a paradox in the search for the magic bullet. There are so many tools, aids, and techniques being proposed that the selection is becoming a problem and the quest for a single solution results in multiple solutions. Many companies are definitely heading toward multiple DBMS environments and systems are being developed with multiple languages (for example, a 4GL and COBOL). It is little wonder that the tools are becoming part of the problem.

D. IBM

- INPUT has often stated that chaos in the marketplace (hardware, software, technology, or the general economy) can only benefit one vendor and that is IBM. The current competitive environment in the use and/or misuse of application development tools definitely falls under the category of chaos and

only IBM will benefit. When users do not know what to do, they naturally turn to IBM, which has the deserved reputation for "making things work" and being around after any technological or economic upheaval.

- The fact of the matter is that IBM's cautious approach in many areas (such as LANs and micro-mainframe links) makes not only good business sense, but good technical sense as well. There are literally times when IBM's best interests and those of its customers coincide. INPUT has stated that IBM's highly centralized strategy during the current SNA/DDP strategic period makes more sense than it has in the past, and that statement is a direct result of the chaos which exists in the DSD environment.

- However, this does not mean that IBM has all the answers, or even the resources to solve the problems existing in the systems development process. In fact, it is not at all certain that IBM either recognizes all of the problems or even wants to solve them. The point is that IBM is in a position to establish the general hardware/software environment in which applications development takes place. This is not meant to be threatening—it is a simple statement of fact which must be recognized if one is to formulate a plan which makes any sense at all.

V CONCLUSIONS AND RECOMMENDATIONS

A. CONCLUSIONS

- INPUT has earlier been concluded that during the SNA/DDP strategic period (the remainder of the 1980s) IBM will place primary software emphasis upon SNA, operating systems, and DBMS. It has also been concluded that IBM's centralized strategy is sound technically in the DSD environment and will be successful in establishing the essential hardware/software environment in which development will take place. At this point, it should be conceded that any applications development tools you employ will, of necessity, have to interface with this environment, including DBMSs.

- It is also concluded that the relational model, and specifically DB2, will serve as the point of interface at all levels in the network hierarchy. In other words, DBMSs employed on mainframes, minicomputers (departmental processors), and intelligent workstations will at some point have to connect with DB2. This is true regardless of whether the DBMS is IBM's or of a competitive variety.

- Therefore, the relational model is of more than academic interest, and at the very least, IBM's DB2 strategy and implementation must be understood. It is important not only for interfacing other DBMSs, but for FGLs as well.

- While IBM's multiple operating system (VM, MVS, UNIX, etc.), DBMS, and language strategy may be inevitable (and even sound in a general technological sense), there are potential quality impacts of significance. The impacts of the DSD environment on systems quality have been detailed (an perhaps belabored) in numerous INPUT reports, but IBM's strategy can have critical impact in one area--hardware/software performance. IBM remains primarily a hardware vendor and has little incentive to be concerned about performance at this level.

- INPUT has concluded that you are going to be spending more for ADTs (that is why the market forecasts were included) and that is only the tip of the iceberg. Use of ADTs (and in the broad sense, operating systems are the primary ADT) add substantially to residual hardware/software costs, and as more complex computer/communications systems are installed the residual cost remains regardless of the institutional value (or use) of the system.

- It is our opinion that there will be increased attention given to these residual costs (hardware/software performance) as the impact of the DSD environment and IBM's strategy develops. This will give rise to the need for vastly improved tools to predict, measure, and control hardware/software performance (costs). Since major hardware and software vendors are disinclined to even recognize the problem, much less do anything about it, you must develop your own quality control program with the help of some of the smaller vendors. Tools to assist you in this area comprise the bulk of the "other" ADT market projected for 1990.

B. RECOMMENDATIONS

- Establish a data base strategy which incorporates the following:

- A plan for the orderly distribution of data at all levels of the network hierarchy. This plan should include standards for DBMS at the specific levels in terms of size, access control, security, and the particular DBMS to be employed.

- On the assumption that communications among the various levels (and between the various DBMSs) will take the form of relational tables, determine the degree of relational purity which is required by your particular organization. In other words, gain some understanding of the current relational controversy and make an informed decision concerning the data base communications vehicle which will be employed across systems.

- Give consideration to the data/information sources which are becoming available on public networks and use them to supplement and complement your data base strategy. The purpose of the evaluation is to establish standards and provide quality data/information (and eventually knowledge) in the most cost-effective manner.

- Establish a quality assurance program for all ADTs and data/information sources based on INPUT's performance levels and with special emphasis upon the often cited quality problems of the DSD environment. Any such program should focus on the residual cost of using and depending upon such tools and data/information sources.

- Recognizing the increasing complexity of measuring, controlling, and predicting hardware/software performance (and the reluctance of major vendors to address these needs), give special attention to this area in any quality assurance program. As applications become distributed over the network hierarchy, even routine cost accounting (and cost recovery) will become more difficult. These problems should be analyzed and resolved now if unpleasant surprises (with severe impact at other performance levels) are to be avoided.

About INPUT

INPUT provides planning information, analysis, and recommendations to managers and executives in the information processing industries. Through market research, technology forecasting, and competitive analysis, INPUT supports client management in making informed decisions. Continuing services are provided to users and vendors of computers, communications, and office products and services.

The company carries out continuous and in-depth research. Working closely with clients on important issues, INPUT's staff members analyze and interpret the research data, then develop recommendations and innovative ideas to meet clients' needs. Clients receive reports, presentations, access to data on which analyses are based, and continuous consulting.

Many of INPUT's professional staff members have nearly 20 years' experience in their areas of specialization. Most have held senior management positions in operations, marketing, or planning. This expertise enables INPUT to supply practical solutions to complex business problems.

Formed in 1974, INPUT has become a leading international planning services firm. Clients include over 100 of the world's largest and most technically advanced companies.

Offices

NORTH AMERICA

Headquarters
1943 Landings Drive
Mountain View, CA 94043
(415) 960-3990
Telex 171407

New York
Parsippany Place Corp. Center
Suite 201
959 Route 46 East
Parsippany, NJ 07054
(201) 299-6999
Telex 134630

Washington, D.C.
11820 Parklawn Drive
Suite 201
Rockville, MD 20852
(301) 231-7350

EUROPE

United Kingdom
INPUT
41 Dover Street
London W1X 3RB
England
01-493-9335
Telex 27113

Italy
Nomos Sistema SRL
20127 Milano
Via Soperga 36
Italy
Milan 284-2850
Telex 321137

Sweden
Athena Konsult AB
Box 22232
S-104 22 Stockholm
Sweden
08-542025
Telex 17041

ASIA

Japan
ODS Corporation
Dai-ni Kuyo Bldg.
5-10-2, Minami-Aoyama
Minato-ku,
Tokyo 107
Japan
(03) 400-7090
Telex 26487

INPUT®
Planning Services For Management

INPUT®
1943 Landings Drive, Mountain View, CA 94043, (415) 960-3990